

WHAT IS CLAIMED IS:

1. In a processing unit including nonvolatile memory for storing executable code, the nonvolatile memory comprising a vital region for storing vital code and a region for storing less vital code, an apparatus for executing an update program for managing updates to the vital region of the nonvolatile memory with update code, the apparatus comprising:

a processor programmed for  
performing a write test by writing test code into the less  
vital region and verifying test status and data, and  
performing an actual write of the update code into the vital  
region if the write test is successful.

2. The apparatus as recited in claim 1, each region of the nonvolatile memory of the processing unit having a version number and likewise each region of the update code having a version number, the apparatus further comprising:

a processor further programmed for  
comparing the version number of the update code and the  
version number of the nonvolatile memory for a given region, and  
executing the update program only if the version number of  
the update code is greater than the nonvolatile memory code version number.

3. The apparatus as recited in claim 2, wherein a location within each vital region is allocated to store the version number, the processor further programmed to access this location to retrieve the version number from the nonvolatile memory.

4. The apparatus as recited in claim 3, the processor further programmed for writing the version number of the update code into the allocated area in that region of nonvolatile memory if the write test is successful.

5. The apparatus as recited in claim 2, the processor further programmed for terminating the update without performing any write operations if the version number of the update code is less than or equal to the nonvolatile memory version number for each vital region.

6. The apparatus as recited in claim 2, the processor further programmed for querying a user whether the update program should proceed if the version number of the update code is less than or equal to the nonvolatile memory version number.

7. The apparatus as recited in claim 1, wherein the vital region is a boot region and the non-vital region is a test region.

8. The apparatus as recited in claim 1, wherein the vital region is a boot region and the less vital region is an application region.

9. The apparatus as recited in claim 8, wherein the test code is equivalent to application code currently stored in the application region of nonvolatile memory.

10. The apparatus as recited in claim 1, wherein the test code is equivalent to the update code.

11. A host bus adapter (HBA) comprising the apparatus of claim 1, the HBA for implementing upper layer protocols (ULPs).

12. The HBA of claim 11, further comprising an Internet Small Computer System Interface (iSCSI) or a fibre channel controller circuit.

13. A host computer comprising the HBA of claim 12.

14. A storage area network (SAN) comprising the host computer of claim 13, wherein an iSCSI or a fibre channel network is coupled to the iSCSI or fibre channel controller circuit and one or more storage devices are coupled to the iSCSI or fibre channel network.

15. In a processing unit including nonvolatile memory for storing executable code, the nonvolatile memory comprising a vital region for storing vital code and a region for storing less vital code, a computer program for managing updates to the vital region of the nonvolatile memory with update code, the computer program being stored on a machine readable medium and executable to perform acts comprising:

performing a write test by writing test code into the less vital region in nonvolatile memory and verifying test status and data; and

performing an actual write of the update code into the vital region in nonvolatile memory, if the write test is successful.

16. The computer program as recited in claim 15, each region of the nonvolatile memory of the processing unit having a version number and the update code having a version number, the computer program further executable to perform acts comprising:

comparing the version number of the update code and the version number of the nonvolatile memory for a given region, and

updating the vital region of the nonvolatile memory only if the version number of the update code is greater than the nonvolatile memory version number.

17. The computer program as recited in claim 16, wherein a location within each vital region is allocated to store the version number, the computer program further executable to perform acts comprising accessing this location to retrieve the version number from both the nonvolatile memory and the update code.

18. The computer program as recited in claim 17, the computer program further executable to perform acts comprising writing the version number of the update code into the allocated area in that region of nonvolatile memory if the write test is successful.

19. The computer program as recited in claim 16, the computer program further executable to perform acts comprising terminating the update program without performing any write operations if the version number of the update code is less than or equal to the nonvolatile memory version number for each vital region.

20. The computer program as recited in claim 16, the computer program further executable to perform acts comprising querying a user whether the update program should proceed if the version number of the update code is less than or equal to the nonvolatile memory code version number.

21. The computer program as recited in claim 15, wherein the vital region is a boot region and the non-vital region is a test region.

22. The computer program as recited in claim 15, wherein the vital region is a boot region and the less vital region is an application region.

23. The computer program as recited in claim 22, wherein the test code is equivalent to application code currently stored in the application region of nonvolatile memory.

24. The computer program as recited in claim 15, wherein the test code is equivalent to the update code.

25. A host bus adapter (HBA) comprising the computer program of claim 15, the HBA for implementing upper layer protocols (ULPs).

26. The HBA of claim 25, further comprising an Internet Small Computer System Interface (iSCSI) or a fibre channel controller circuit.

27. A host computer comprising the HBA of claim 26.

28. A storage area network (SAN) comprising the host computer of claim 27, wherein an iSCSI or fibre channel network is coupled to the iSCSI or fiber channel controller circuit and one or more storage devices are coupled to the iSCSI or fibre channel network.

29. In a processing unit including nonvolatile memory for storing executable code, the nonvolatile memory comprising a vital region for storing vital code and a region for storing less vital code, a method for executing an update program for managing updates to the vital region of the nonvolatile memory with update code, the method comprising:

performing a write test by writing test code into the less vital region and verifying test status and data, and

performing an actual write of the update code into the vital region if the write test is successful.

30. The method as recited in claim 29, each region of the nonvolatile memory of the processing unit having a version number and the update code having a version number, the method further comprising:

comparing the version number of the update code and the version number of the nonvolatile memory for a given region, and

executing the update program only if the version number of the update code is greater than the nonvolatile memory code version number.

31. The method as recited in claim 30, wherein a location within each vital region is allocated to store the version number, the method further comprising accessing this location to retrieve the version number from the nonvolatile memory.

32. The method as recited in claim 31, further comprising writing the update code version number of the update code into the allocated area in that region of nonvolatile memory if the write test is successful.

33. The method as recited in claim 30, further comprising terminating the update without performing any write operations if the version number of the update code is less than or equal to the nonvolatile memory code version number for each vital region.

34. The method as recited in claim 30, further comprising querying a user whether the update program should proceed if the version number of the update code is less than or equal to the nonvolatile memory code version number.

35. The method as recited in claim 29, wherein the vital region is a boot region and the less vital region is a test region.

36. The method as recited in claim 29, wherein the vital region is a boot region and the less vital region is an application region.

37. The method as recited in claim 36, wherein the test code is equivalent to application code currently stored in the application region.

38. The method as recited in claim 29, wherein the test code is equivalent to the update code.